

## Desain Sistem Kontrol Ketinggian Air Bendungan Menggunakan Metode Optimasi *Genetic Algorithm*

Arif Setiawan

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya  
e-mail : arif.17050874065@mhs.unesa.ac.id

Muhamad Syariffuddin Zuhrie, I Gusti Putu Asto Buditjahjanto, Lilik Anifah

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya  
e-mail : zuhrie@unesa.ac.id, asto@unesa.ac.id, lilikanifah@unesa.ac.id

### Abstrak

Bendungan adalah suatu wadah untuk menampung maupun menahan air. Manfaat dari bendungan sangat banyak sekali, salah satunya yaitu sebagai pengendali banjir, sehingga tidak terjadi bencana banjir dalam suatu wilayah tersebut. Dengan banyaknya manfaat bendungan, maka perlu adanya sistem kontrol yang dapat mengontrol ketinggian air bendungan agar ketinggian air tersebut tetap berada pada keadaan yang stabil. Pada penelitian ini penulis menerapkan metode optimasi *Genetic Algorithm* sebagai pengontrol ketinggian air pada bendungan untuk mendapatkan hasil respon sesuai dengan *set point* yang diinginkan dan dapat mengurangi *overshoot* dan *error* pada sistem tersebut. Metode optimasi Algoritma Genetika bertujuan mencari nilai  $K_p$ ,  $K_i$ , dan  $K_d$  dengan optimal yang mana nilai tersebut akan diterapkan pada kontrol PID. Setelah melakukan penelitian, diperoleh parameter PID dengan menggunakan optimasi *Genetic Algorithm* yakni  $K_p = 4.75$ ,  $K_i = 0.069$ ,  $K_d = 59.666$ . Setelah mendapatkan parameter tersebut dilakukan pengujian secara simulasi dengan menggunakan *software* matlab yang mana didapatkan respon yang cukup baik dengan tidak adanya *overshoot* dan *error* pada sistem tersebut. Respon yang baik terdapat pada *set point* 12 cm dengan nilai *rise time* ( $t_r$ ) = 23.4905 s, *settling time* ( $t_s$ ) = 42.047 s, *error steady state* = 0%, *overshoot* = 0%. Meskipun pada metode *fuzzy* memiliki nilai *rise time* ( $t_r$ ) yang lebih baik, namun dengan metode optimasi *Genetic Algorithm* mampu memperbaiki nilai *settling time* serta memperkecil nilai *overshoot* dan *error* pada sistem tersebut.

**Kata Kunci:** *Genetic Algorithm*, Kontrol Ketinggian Air, PID.

### Abstract

A dam is a container to hold water. The benefits of dams are very many, one of which is as a flood control, so that no flood disaster in an area. With many benefits a dam, it is necessary to have a control system that can control the water level of the dam so that the water level remains in a stable state. In this study, the authors applied the Genetic Algorithm optimization method to control the water level in the dam to obtain response results according to the desired set point and to reduce overshoot and errors in the system. The Genetic Algorithm optimization method aims to find the optimal  $K_p$ ,  $K_i$ , and  $K_d$  values which these values will be applied to PID control. After conducting the research, the PID parameters were obtained using the Genetic Algorithm optimization, namely  $K_p = 4.75$ ,  $K_i = 0.069$ ,  $K_d = 59.666$ . After obtaining these parameters, a simulation test was carried out using the matlab software which obtained a good response in the absence of overshoot and errors in the system. A good response is found at the 12 cm set point with the value rise time ( $t_r$ ) = 23.4905 s, settling time ( $t_s$ ) = 42.047 s, steady state error = 0%, overshoot = 0%. Even though the fuzzy method has a better rise time ( $t_r$ ) value, the Genetic Algorithm optimization method is able to improve the settling time value and reduce the overshoot and error values in the system.

**Keywords:** Genetic Algorithm, PID, Water Level Control.

### PENDAHULUAN

Bendungan adalah suatu wadah untuk menampung maupun menahan air, dimana air tersebut dapat dimanfaatkan untuk berbagai tujuan. Manfaat dari bendungan sangat banyak sekali, salah satunya yaitu sebagai pengendali banjir, sehingga tidak terjadi bencana banjir dalam suatu wilayah tersebut. Selain itu bendungan juga dapat digunakan sebagai penyediaan air untuk irigasi, air baku, air minum, atau dapat digunakan sebagai

pembangkit listrik serta pengembangan lokasi kawasan wisata (Sunardi dan Rara, 2014).

Bencana banjir menjadi suatu problematika yang kerap terjadi di Negara Indonesia, terlebih lagi di daerah dataran rendah. Bencana banjir dapat terjadi karena beberapa faktor, yang mana diantaranya yaitu; banyaknya pembangunan dilakukan tanpa memperdulikan lahan hijau, minimnya kesadaran masyarakat tentang pentingnya merawat aliran air yang terdapat pada area waduk atau bendungan, masih banyaknya penduduk membuang

sampah tidak pada tempatnya, mengakibatkan terjadi penumpukan sampah serta penyumbatan sungai (Alfatah, 2016).

Dengan banyaknya manfaat bendungan, maka diperlukan sebuah sistem kontrol yang bisa mengontrol level air bendungan supaya ketinggian air tersebut selalu berada pada keadaan yang stabil. Sistem ini diperlukan untuk mengatasi masalah terhadap curah hujan yang tidak dapat diprediksi pada masing-masing wilayah. Di Indonesia sendiri masih banyak bendungan yang menggunakan sistem manual dalam pengoperasiannya, sehingga masih terdapat kendala-kendala yang kita tidak inginkan.

Pada penelitian sebelumnya yang telah dilakukan Sudaryoto pada tahun 2019 dengan judul “Rancang Bangun Sistem Kontrol Ketinggian Air Bendungan Berbasis *Fuzzy Logic Controller*” menghasilkan respon sistem yang dimana pada *set point* 10 cm didapatkan nilai waktu tunak ( $t_s$ ) = 111.731 s, waktu naik ( $t_r$ ) = 16.543 s, *overshoot* = 17.34% dan *Ess* = 2.6%, pada *set point* 12 cm didapatkan nilai waktu tunak ( $t_s$ ) = 98.765 s, waktu naik ( $t_r$ ) = 18.378 s, *overshoot* = 6.4% dan *Ess* = 2.84%, dan pada *set point* 14 cm didapatkan nilai waktu tunak ( $t_s$ ) = 93.678 s, waktu naik ( $t_r$ ) = 15.66 s, *overshoot* = 11.29% dan *Ess* = 3%.

Pada penelitian ini penulis mengembangkan suatu sistem kendali dengan metode optimasi Algoritma Genetika, yang mana dengan tujuan dapat menghasilkan serta dapat melihat hasil dari percobaan suatu sistem kontrol level air bendungan saat memakai metode optimasi *Genetic Algorithm*. Sehingga kita diharapkan dapat memperkecil *overshoot* dan *error* yang ada pada respon sistem tersebut.

## METODE

### Pendekatan Penelitian

Penelitian ini disusun berdasarkan pada metode pendekatan penelitian kuantitatif. Menggunakan pendekatan penelitian kuantitatif dikarenakan pada percobaan kali ini, data yang diteliti sebuah angka. Data inilah yang akan dikumpulkan serta yang akan di analisa (Kasiram, 2008). Pengambilan data akan dilakukan beberapa kali guna memastikan kinerja dari suatu sistem tersebut.

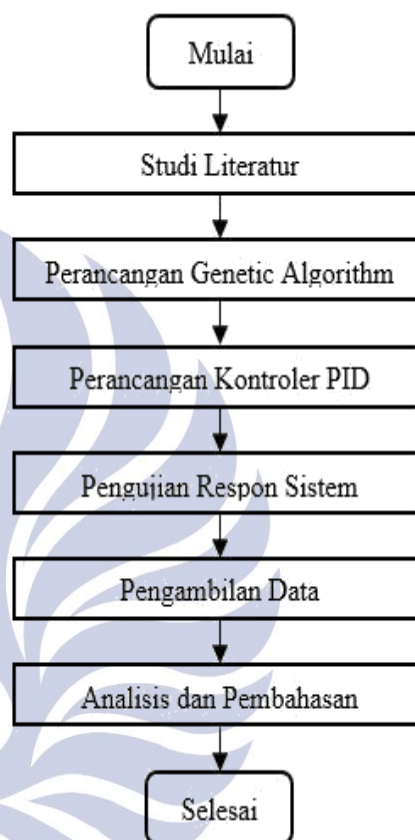
### Instrumen Pengumpulan Data

Instrumen penelitian merupakan sebuah alat bantu yang mana berfungsi sebagai pengumpul data yang diperlukan (Nasution, 2016). Pada penelitian ini penulis menggunakan *software* Matlab 2016a yang telah terinstal pada laptop. Matlab 2016a disini dimanfaatkan untuk mencari nilai dari  $K_p$ ,  $K_i$ , dan  $K_d$  dengan cara metode optimasi Algoritma Genetika yang mana nantinya akan dimasukkan dalam kontroler PID. Selain itu Matlab 2016a digunakan untuk

menjalankan simulasi dan menampilkan respon dari sistem kontrol ketinggian air bendungan.

### Rancangan Penelitian

Rancangan penelitian dapat dilihat dari *flowchart* Gambar 1.

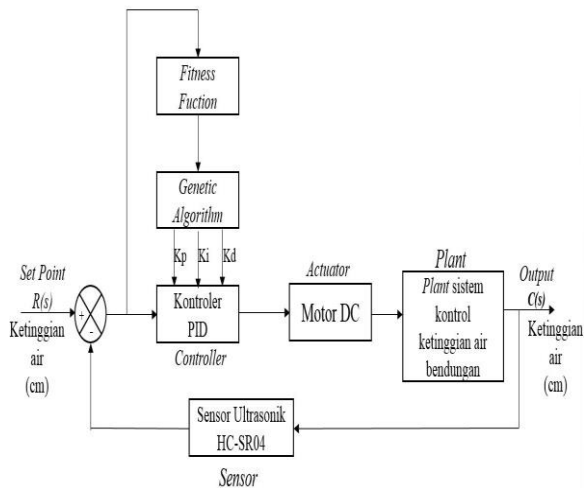


Gambar 1. Skema *Flowchart* Rancangan Penelitian

Berdasarkan Gambar 1 penyusunan penelitian ini pertama kali dimulai dari studi literatur dari berbagai referensi seperti jurnal, skripsi, dan *ebook*. Setelah itu membuat perancangan *Genetic Algorithm* untuk menentukan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  menggunakan *software* matlab 2016a. Kemudian membuat perancangan kontroler PID dan melakukan pengujian simulasi menggunakan *software* matlab 2016a. Setelah simulasi dilakukan maka perlu adanya pengambilan data pada simulasi. Langkah terakhir yaitu melakukan analisis pada data yang sudah didapat.

### Desain Sistem

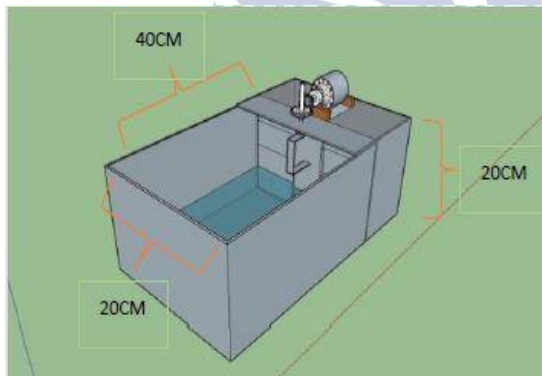
Desain diagram blok sistem kontrol ketinggian air bendungan dengan metode optimasi *Genetic Algorithm* ditunjukkan pada Gambar 2.



Gambar 2. Diagram Blok Sistem

### Pemodelan Dinamik Bendungan

Desain pada sistem bendungan ini menggunakan sistem pintu yang mana akan digerakan oleh sebuah *actuator* yaitu motor DC. Rancang bangun dapat dilihat seperti pada Gambar 3.

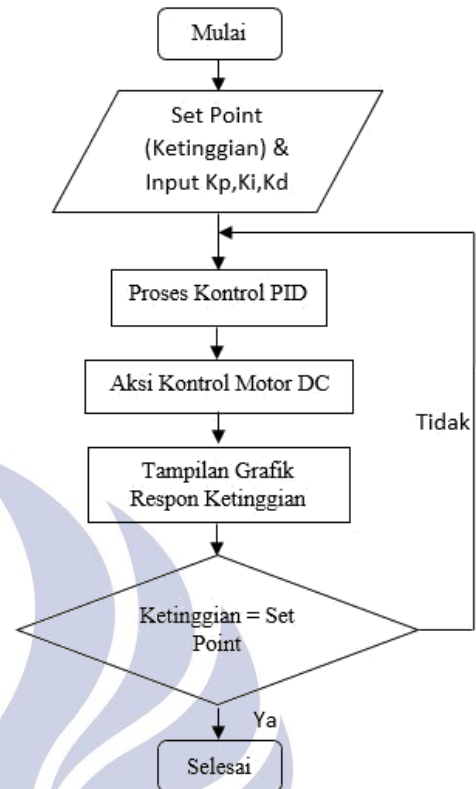


Gambar 3. Rancang Bangun Bendungan  
(Sumber: Wahyono, 2019)

Input disini berupa ketinggian air yang mana akan dikontrol oleh kontroler PID dengan metode optimasi Algoritma Genetika sehingga akan membuat motor DC menggerakkan gerbang pada saat mencapai *set point* yang telah diatur. Sensor ultrasonik akan mengirim sinyal umpan balik berupa ketinggian air secara berkelanjutan sehingga dapat mencapai ketinggian air yang sesuai dengan *set point* yang telah ditentukan.

### Perancangan Software

Diagram alir perancangan *software* pada sistem kendali ketinggian air bendungan dapat dilihat pada *flowchart* Gambar 4.



Gambar 4. Skema *Flowchart* Perancangan *Software*

Berdasarkan Gambar 4. Perancangan *software* dimulai dengan memasukkan nilai *set point* serta penginput nilai  $K_p$ ,  $K_i$ , dan  $K_d$  pada *PID controller* yang mana telah didapat melalui metode *Genetic Algorithm*, proses akan dilanjutkan pengontrolan *PID* yang mana akan memberikan aksi motor DC sebagai penggerak pintu bendungan serta akan ditampilkan grafik respon ketinggiannya, proses akan berhenti jika ketinggian mencapai atau sama dengan *set point* dan akan berlanjut apabila ketinggian belum mencapai *set point*.

### Karakteristik Sistem Orde Dua

Respon orde-2 berfungsi untuk mendapatkan sebuah respon sistem kontrol seperti step input, ramp input, dan impulse input yang mana memiliki ciri dari orde-2 (Abdillah, 2021).

$$\frac{C(s)}{R(s)} = \frac{K}{J s^2 + B s + K} \quad (1)$$

Persamaan (1) yang mana merupakan fungsi alih orde-2 dapat juga ditulis sebagai berikut;

$$\frac{C(s)}{R(s)} = \frac{\frac{K}{J}}{[s + \frac{B}{2J}] + \sqrt{(\frac{B}{2J})^2 - \frac{K}{J}}}[s + \frac{B}{2J} - \sqrt{(\frac{B}{2J})^2 - \frac{K}{J}}] \quad (2)$$



Pada analisis respon transien, dapat ditulis sebagai berikut;

$$\frac{K}{J} = \omega_n^2 \quad (3)$$

$$\frac{B}{J} = 2\zeta\omega_n = 2\sigma \quad (4)$$

Keterangan :

$\sigma$  = Attenuation

$\omega_n$  = The Undamped Natural Frequency

$\zeta$  = The Damping Ratio of System

$K$  = Gain

$B$  = Actual Damping

$J$  = Momen Inersia

Rasio *Damping* merupakan perbandingan dari *damping* aktual  $B$  dengan *damping critical*  $B_c = 2\sqrt{JK}$  maka dapat ditulis sebagai berikut ;

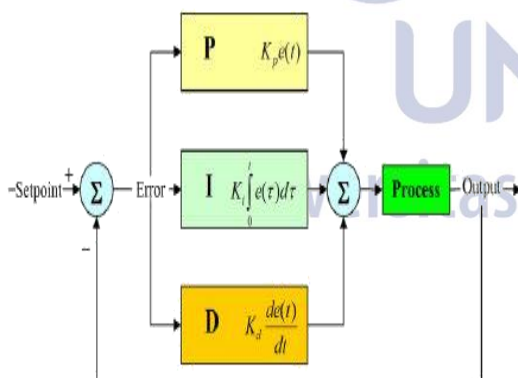
$$\zeta = \frac{B}{B_c} = \frac{B}{2\sqrt{JK}} \quad (5)$$

Dengan ketentuan dari  $\zeta$  dan  $\omega_n$ , transfer fungsi *close loop* dapat ditulis dengan rumus ;

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (6)$$

### Kontroler PID

Kontrol PID merupakan sistem pengendali yang umum digunakan dalam dunia industri, yang mana sekitar 90% dari peralatan yang ada di industri menggunakan pengendali PID (Wisnu, dkk., 2016). Blok diagram kontrol PID dapat dilihat seperti pada Gambar 5.



Gambar 5. Blok Diagram Pengendali PID  
(Sumber: Wisnu, dkk., 2016)

Pada kontrol PID memiliki nilai output  $u(t)$  dari penjumlahan tiga variabel yang mana dapat dilihat dari persamaan berikut :

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (7)$$

Keterangan :

$u(t)$  = Sinyal Output Kontroler

$K_p$  = Proportional gain

$T_i$  = Waktu integral

$T_d$  = Waktu derivative

$e(t)$  = Sinyal error yang dihasilkan

Pada kendali PID terdapat *set point* yang mana dijadikan sebagai penginput besaran nilai yang diinginkan menghasilkan suatu keluaran seperti yang telah ditetapkan (Ogata, 1985).

Pada kntroler PID sendiri terdapat parameter yang harus diatur agar menghasilkan sinyal keluaran sistem sesuai masukan yang telah ditetapkan (Ali, 2004). Tabel 1. merupakan tanggapan respon pada kontroler PID dari masing-masing parameter.

Tabel 1. Respon Kendali PID

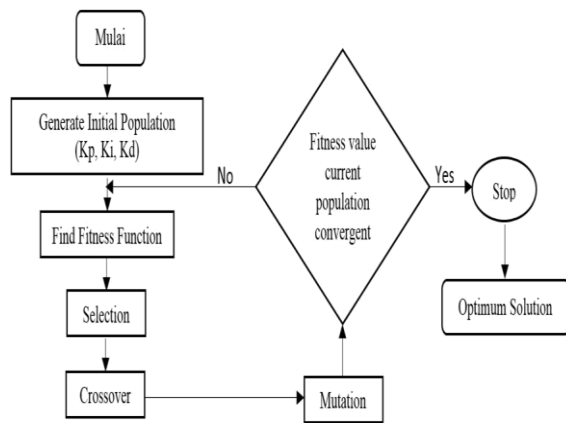
Close Loop	Rise Time	Overshoot	Settling Time	SS Error
$K_p$	Menurun	Meningkat	Perubahan Kecil	Menurun
$K_i$	Menurun	Meningkat	Meningkat	Hilang
$K_d$	Perubahan kecil	Menurun	Menurun	Perubahan Kecil

### Genetic Algorithm

*Genetic Algorithm* merupakan suatu metode pencarian acak berfungsi sebagai penyelesaian sistem nonlinier dan dapat mengoptimalkan masalah yang kompleks. Dalam *Genetic Algorithm* menggunakan aturan transisi probabilistik. Dalam setiap iterasi dari *algorithm* tersebut diistilahkan sebagai *generation*. Evolusi dari sebuah solusi akan disimulasikan melalui *fitness fuction* dan operator *genetic* seperti *reproduction*, *crossover*, dan *mutation* (Jayachitra dan Vinodha, 2014).

Algoritma Genetika sering digunakan pada pemecahan permasalahan yang dimana mempunyai banyak solusi dan akan menemukan solusi yang optimal dari beberapa solusi tersebut. Aplikasi Algoritma Genetika sering digunakan sebagai simulasi dengan komputer guna mendapatkan solusi yang paling optimal.

Dari sebagian operator menggunakan *crossover* dan *mutation* untuk menciptakan populasi yang baru dari hasil populasi yang terdahulu dan terus akan terulang hingga kriteria yang ditentukan oleh *algorithm* berhenti (Abdillah, 2021). Skema *Genetic Algorithm* dapat ditunjukkan pada Gambar 6.



Gambar 6. Flowchart Genetic Algorithm

*Genetic Algorithm* memiliki beberapa tahapan proses yang mana dapat ditunjukkan pada Gambar 6. Diantaranya yaitu :

1. *Generate Initial Population*  
Tahap ini merupakan tahap untuk menentukan beberapa individu dari populasi yang mana mempresentasikan sebagai solusi dalam permasalahan.
2. *Find fitness fuction*  
Pada tahap ini *fitness fuction* menghitung berapa skor *fitnes* pada setiap individu. Dalam perhitungan ini akan diketahui apakah individu pada tiap kromosom mempunyai kualitas yang baik atau tidak dari yang lainnya.
3. *Selection*  
Tahap dimana mencari individu yang mempunyai kualitas yang baik dan akan mewariskan gen individu pada generasi selanjutnya.
4. *Crossover*  
Tahapan dimana akan dilakukan rekombinasi untuk mendapatkan individu yang lebih bagus.
5. *Mutation*  
Pada tahapan ini dilakukanya mutasi dari beberapa gen dengan kemungkinan acak yang rendah pada keturunan baru yang telah terbentuk

## HASIL DAN PEMBAHASAN

Pengambilan data, perancangan serta analisis untuk dapat mengetahui respon sistem dalam penelitian dilakukan dengan menggunakan software MATLAB R2016a.

### Pemodelan Sistem

Pemodelan sistem pada penelitian ini penulis menggunakan pemodelan sistem seperti yang digunakan oleh peneliti sebelumnya (Sudaryoto, 2019) yang mana

menggunakan pendekatan karakteristik sistem orde 2 agar dapat memperoleh sebuah fungsi transfer. Pada penelitian ini menggunakan *set point* 10 cm dengan nilai fungsi transfer dari nilai acuan jurnal sebelumnya sebagai berikut:

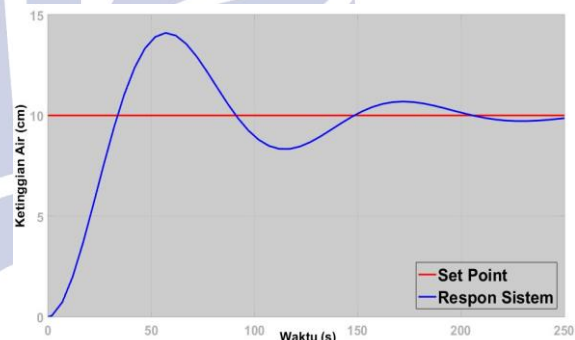
$$\frac{C(s)}{R(s)} = \frac{0.003249}{s^2 + 0.031122s + 0.003249} \quad (8)$$

Setelah didapatkan transfer fungsi, maka kita akan melihat bagaimana respon yang didapat dari sistem *open loop* tanpa kontroler PID dengan cara mensimulasikan transfer fungsi tersebut ke diagram blok *open loop* sistem yang mana dapat dilihat pada gambar 7.



Gambar 7. Simulink Open Loop

Hasil respon *open loop* dapat ditunjukkan pada Gambar 8.



Gambar 8. Grafik Respon Open Loop

Berdasarkan Gambar 8. Menunjukkan bahwa respon keluaran yang dihasilkan sistem *open loop* tanpa kontroler PID masih kurang stabil dimana masih terdapatnya overshoot serta untuk mencapai *set point* (ketinggian air) masih membutuhkan waktu yang cukup lama. Sehingga sistem tersebut memerlukan sebuah kontroler agar mendapatkan hasil respon keluaran yang stabil.

### Perancangan Kontroler PID

Metode optimasi *Genetic Algorithm* digunakan sebagai pencarian sebuah nilai Kp, Ki, dan Kd pada kendali PID yang berfungsi sebagai kontroler untuk mengontrol ketinggian air pada bendungan. Setelah mengetahui bagaimana respon dari *open loop* kemudian kita akan menentukan nilai Kp, Ki, Kd dengan membuat *script fitness function* seperti pada Gambar 9.

```

function [J] = pid_optim(x)

s = tf('s');

plant = 0.003249 / (s^2 + 0.031122*s + 0.003249);

Kp = x(1)
Ki = x(2)
Kd = x(3)

controller = Kp + Ki/s + Kd*s;

%step(feedback(controller*plant,1));

dt = 0.9;
t = 0:dt:1;

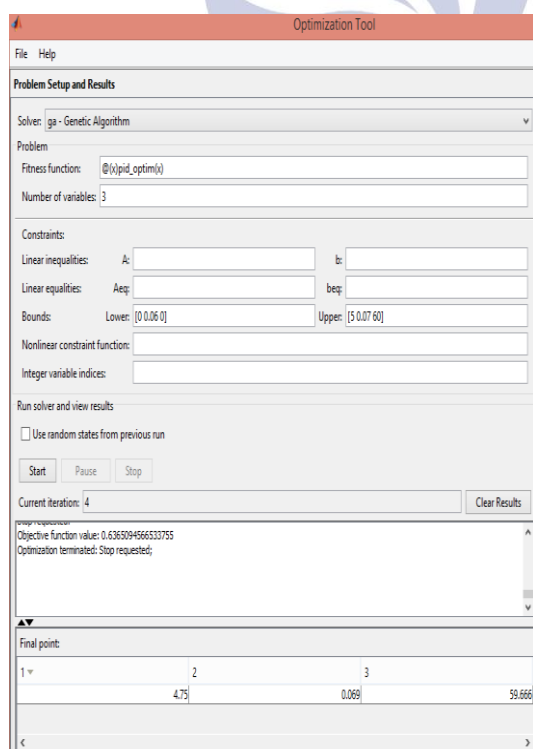
e = 1 - step(feedback(controller*plant,1),t);

J = sum(t'.*abs(e)*dt);

```

Gambar 9. Script Fitness Function

Setelah selesai membuat *script function* seperti Gambar 9, maka kita akan mengaplikasikan *script function* tersebut kedalam fitur *optimization*. Pada fitur *optimization* kita akan memilih *Genetic Algorithm* pada menu *solver* seperti yang dapat dilihat pada Gambar 10.



Gambar 10. Fitur Optimization

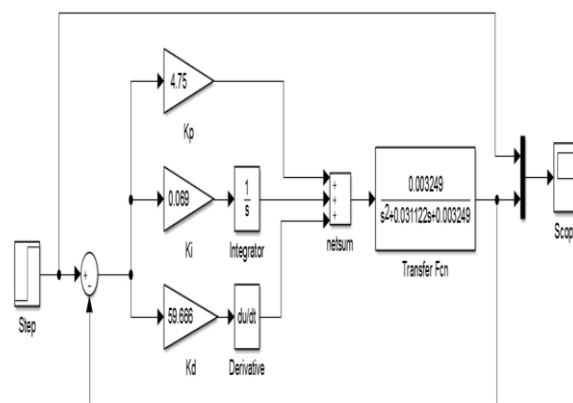
Variabel yang kita masukan berjumlah 3 variabel yaitu  $K_p$ ,  $K_i$ , dan  $K_d$ . Setelah itu kita harus menentukan batas bawah dan batas atas pada menu *bounds* untuk

mendapatkan hasil yang optimal. Ketika semua menu yang dibutuhkan telah terisi semua, maka kita dapat menjalankannya dengan menekan tombol *start* dan akan bekerja dengan cara *looping* secara otomatis. Untuk menghentikan kerja sistem tersebut dapat dilakukan oleh *user* dengan menekan tombol *stop*.

Pada percobaan yang telah dilakukan, diperlukan beberapa kali percobaan sampai mencapai 4 iterasi dan didapatkan nilai  $K_p = 4.75$ ,  $K_i = 0.069$ ,  $K_d = 59.666$ . Setelah mendapatkan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  maka kita akan melakukan simulasi pada rangkaian *close loop* PID *controller* seperti yang ditunjukkan pada Gambar 11. Pada pengujian kali ini, simulasi akan dilakukan sebanyak 3 kali dengan menggunakan *set point* 10 cm, *set point* 12 cm, dan *set point* 14 cm dengan masing-masing waktu simulasi 250 detik.

### Perancangan Simulasi pada Matlab

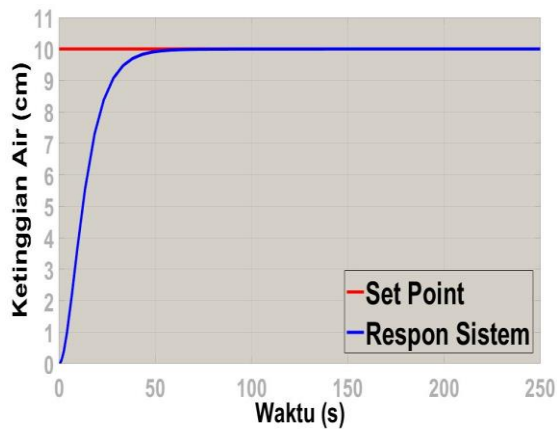
Pada Matlab terdapat program *simulink* yang mana berfungsi untuk melakukan simulasi terhadap percobaan yang akan dilakukan. Pada percobaan kali ini akan melakukan suatu pengujian respon  $K_p$ ,  $K_i$ , dan  $K_d$  yang telah didapat ke diagram blok rangkaian *close loop* dengan *set point* yang telah ditentukan. Diagram blok pada *simulink* dapat dilihat pada Gambar 11.



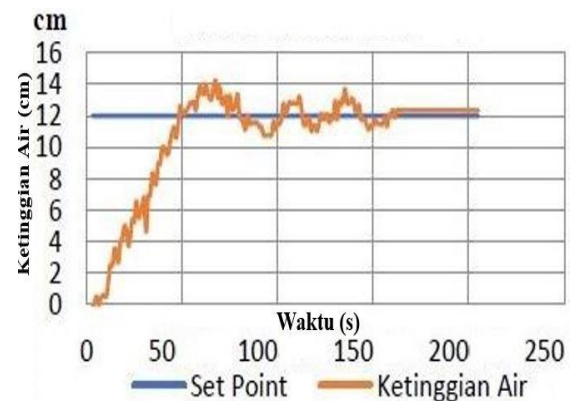
Gambar 11. Rangkaian Close Loop PID Controller

### Pengujian Respon Sistem

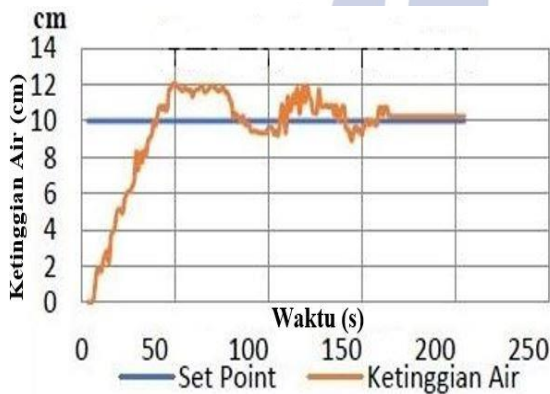
Pengujian respon akan dilakukan sebanyak 3 kali dengan menggunakan *set point* 10 cm, *set point* 12 cm, dan *set point* 14 cm. Hasil respon sistem yang dihasilkan saat menggunakan kontroler PID dengan metode *Genetic Algorithm* pada rangkaian *close loop* ditunjukkan pada Gambar 12, Gambar 14, dan Gambar 16. Sedangkan Gambar 13, Gambar 15, dan Gambar 17 menunjukkan hasil respon keluaran (*real plant*) pada jurnal acuan dimana menggunakan *fuzzy* sebagai kontrolernya.



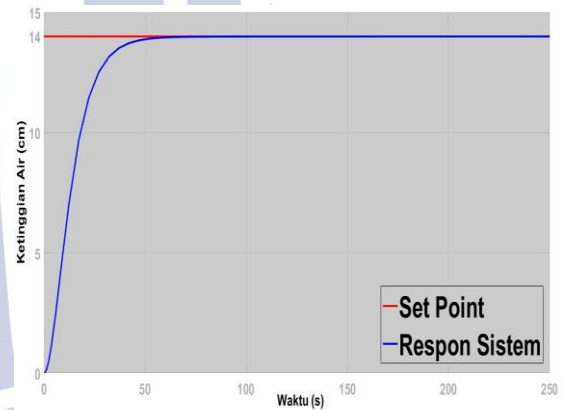
Gambar 12. Respon *close loop* PID controller pada *set point* 10 cm dengan metode *Genetic Algorithm*



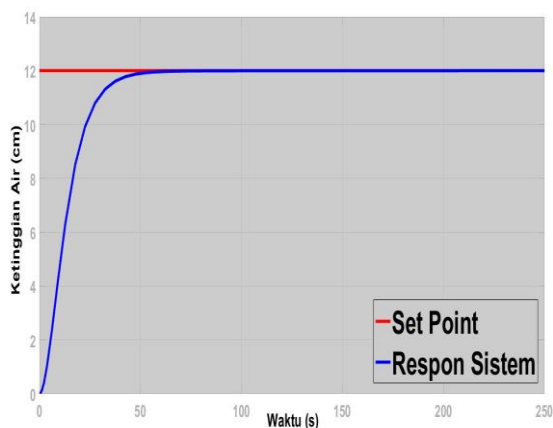
Gambar 15. Respon keluaran (*real plant*) jurnal acuan dengan metode *fuzzy* pada *set point* 12 cm (Sumber: Sudaryoto, 2019)



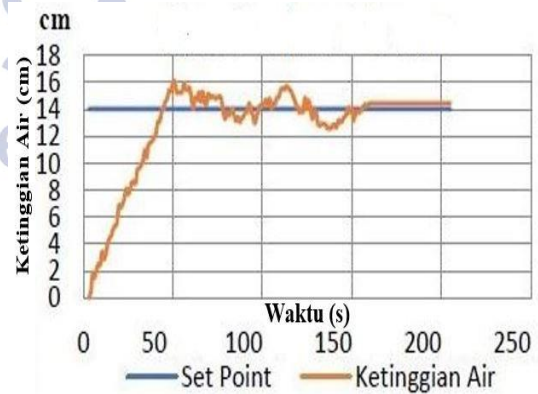
Gambar 13. Respon keluaran (*real plant*) jurnal acuan dengan metode *fuzzy* pada *set point* 10 cm (Sumber: Sudaryoto, 2019)



Gambar 16. Respon *close loop* PID controller pada *set point* 14 cm dengan metode *Genetic Algorithm*



Gambar 14. Respon *close loop* PID controller pada *set point* 12 cm dengan metode *Genetic Algorithm*



Gambar 17. Respon keluaran (*real plant*) jurnal acuan dengan metode *fuzzy* pada *set point* 14 cm (Sumber: Sudaryoto, 2019)



Dari hasil simulasi respon sistem menggunakan PID kontroler pada rangkaian *close loop* pada Gambar 12, Gambar 14, dan Gambar 16 menunjukkan bahwa respon yang di hasilkan cukup baik dimana tidak memiliki *overshoot* dibandingkan dengan respon yang dihasilkan pada rangkaian *open loop* tanpa PID kontroler yang mana masih terdapat *overshoot* seperti ditunjukkan pada Gambar 8.

Respon yang dihasilkan pada saat *set point* 10 cm, 12 cm, dan 14 cm dengan metode *Genetic Algorithm* lebih stabil yang mana memiliki *overshoot* yang lebih kecil dari pada respon keluaran (*real plant*) pada jurnal acuan dengan metode *fuzzy* pada saat *set point* yang sama, yang mana dapat ditunjukkan grafik respon pada Gambar 12, 13, 14, 15, 16, dan 17.

Metode optimasi Algoritma Genetika disini berperan penting untuk menghasilkan sebuah nilai  $K_p$ ,  $K_i$ , dan  $K_d$  yang cukup optimal, sehingga *PID Controller* dapat mengontrol ketinggian air bendungan dengan cukup baik. Dari percobaan simulasi yang dilakukan pada 3 *set point*, dapat diketahui bahwa sistem tersebut mampu melakukan kontrol ketinggian air dengan cukup baik. Hasil analisa dari ketiga respon sistem dengan *set point* yang berbeda akan ditunjukkan pada Tabel 2.

Tabel 2. Hasil Simulasi Respon Sistem

<i>Set point</i>	10 cm	12 cm	14cm	Satuan
<i>Rise Time</i> (tr)	23.582	23.4905	23.6564	<i>Second</i>
<i>Settling Time</i> (ts)	42.138	42.047	41.9471	<i>Second</i>
<i>Error Steady State</i>	0	0	0	%
<i>Overshoot</i>	0	0	0	%

Tabel 3. Hasil Perbandingan dengan Jurnal Acuan

Parameter	Fuzzy	Genetic Algorithm
<i>Rise Time</i> (tr)	18.378 s	23.4905 s
<i>Settling Time</i> (ts)	98.765 s	42.047 s
<i>Error Steady State</i>	6.4%	0%
<i>Overshoot</i>	2.84%	0%

Hasil analisis pada Tabel 2. Pada saat menggunakan optimasi *Genetic Algorithm* menunjukkan bahwa pada *set point* 10 cm memiliki nilai waktu naik (tr) = 23.582 s dan waktu tunak (ts) = 42.138 s. pada *set point* 12 cm memiliki nilai waktu naik (*rise time*) paling cepat dengan nilai 23.4905 s dan memiliki nilai waktu tunak (ts) = 42.047 s, dan pada *set point* 14 cm memiliki waktu tunak (*settling*

*time*) paling cepat dengan nilai 41.947 s dan memiliki nilai waktu naik (tr) = 23.6564 s. Untuk nilai *error steady state* serta *overshoot* pada ketiga *set point* (10 cm, 12 cm, dan 14 cm) memiliki nilai yang sama, yaitu 0%.

Pada pengujian ini dengan Metode optimasi *Genetic Algorithm* yang mana didapatkan nilai  $K_p = 4.75$ ,  $K_i = 0.069$ ,  $K_d = 59,666$  memiliki respon sistem yang lebih baik daripada menggunakan metode *fuzzy* dengan dibuktikan hasil data pada Tabel 3. Dimana saat menggunakan metode optimasi *Genetic Algorithm* respon sistem memiliki *overshoot* dan *error* yang lebih kecil daripada menggunakan metode *fuzzy*.

## PENUTUP

### Simpulan

Setelah mendapatkan hasil penelitian, didapatkan sebuah kesimpulan bahwa metode optimasi Algoritma Genetika dapat mengontrol ketinggian air bendungan dengan cukup baik karena dapat memberikan suatu respon yang sama dengan *set point* yang telah ditetapkan. Dengan optimasi *Genetic Algorithm* didapatkan nilai  $K_p = 4.75$ ,  $K_i = 0.069$ ,  $K_d = 59,666$  yang mana pada respon yang dihasilkan tidak terdapat *overshoot*.

Pada metode *fuzzy* memiliki *rise time* lebih cepat dibandingkan saat menggunakan metode optimasi *Genetic Algorithm*, namun dengan metode *Genetic Algorithm* mampu memperbaiki nilai *settling time* serta memperkecil *overshoot* dan *error* pada sistem tersebut. Hal ini dapat dibuktikan dari hasil data simulasi yang telah didapat yang mana didapat nilai *settling time* (ts) = 42.047 s, *error steady state* = 0%, *overshoot* = 0% yang terjadi pada *set point* 12 cm.

### Saran

Dalam hali ini penulis dapat memberikan saran kepada peneliti selanjutnya untuk mencari model matematika baru dengan parameter-parameter yang telah ditentukan atau dapat menggunakan metode lainnya seperti *Linear Quadratic Regulator control* (LQR), *Adaptive Neuro Fuzzy Inference System* (ANFIS) sehingga mendapatkan suatu respon yang lebih baik lagi.

## DAFTAR PUSTAKA

Abdillah, Riztan Hanif. 2021. "Perancangan Sistem Kontrol Vibrasi pada Robot Single Link Flexible Joint Manipulator dengan PID Tuning Genetic Algorithm". Jurnal teknik Elektro, Universitas Surabaya, Volume 10, No.01, 2021.



Alfatah, Muhammad Rosyid. 2016. "*Prototype Sistem Buka Tutup Otomatis Pada Pintu Air Bendungan untuk Mengatur Ketinggian Air Berbasis Arduino*". Jurnal Skripsi. Surakarta: Universitas Muhammadiyah Surakarta.

Ali, Muhammad. 2004. "*Pembelajaran Perancangan Sistem Kontrol PID dengan Software Matlab*". Jurnal Edukasi. Universitas Negeri Yogyakarta, Vol.1, No.1, Oktober 2004.

Jayachitra, A. dan R. Vinodha. 2014. "*Genetic Algorithm Based PID Controller Tuning Approach for Continuous Stirred Tank Reactor*". Jurnal Penelitian. India: Annamalai University.

Kasiram, Moh. 2008. *Metodologi Penelitian Kualitatif dan Kuantitatif*. Malang: UIN Maliki Press.

Nasution, Hamni Fadlilah. 2016. "*Instrumen Penelitian dan Urgensinya dalam Penelitian Kuantitatif*". Jurnal Universitas IAIN Padangsidimpuan.

Ogata, Katsuhiko. 1985. *Teknik Kontrol Automatik* jilid 1. Terjemahan Edi Laksono. Jakarta: Erlangga.

Sudaryoto, Satria Bagaskara. 2019. "*Rancang Bangun Sistem Kontrol Ketinggian Air Bendungan Berbasis Fuzzy Logic Controller*". Jurnal Skripsi. Surabaya: Program Studi Elektro, Fakultas Teknik, Universitas Negeri Surabaya.

Sunardi dan Rara Sri Artati. 2014. "*Aplikasi Metode Fuzzy Sugeno untuk Sistem Informasi Ketinggian Air dan Ketinggian Pintu Air Suatu Bendungan*". Jurnal Teknologi Informasi, Universitas Stikubank Semarang, Volume 19, No.2, Juni 2014.

Wahyono, Hendrawan. 2019. "*Rancang Bangun Sistem Kontrol Ketinggian Air Bendungan Menggunakan Metode PID*". Jurnal Skripsi. Surabaya: Program Studi Elektro, Fakultas Teknik, Universitas Negeri Surabaya.

Wisnu, Danu., Wahyudi, Arif dan Nurhadi, Hendro. 2016. "*Perancangan Sistem Kontrol PID untuk Pengendali Sumbu Azimuth Turret pada Turret-gun Kaliber 20mm*". Jurnal Teknik ITS, Institut Teknologi Sepuluh Nopember, Vol.5, No.2, 2016.